ملاحظة :
في المحاضرة السابقة في تمرين while لو اكتب برنامج !n
while (i<n);

المصفوفات :
١ المصفوفة ذات البعد الواحد : وصيغتها
نوع المتغيرات اسم المصفوفة [عدد العناصر] ;
int A[5] ;
مثال : A[0] , A[1] , A[2] , A[3] , A[4]
للوصول إلى عناصر المصفوفة نكتب [دليل] A
لإدخال عناصر وطباعة عناصر المصفوفة نستخدم حلقة for
- من أخطاء المصفوفات عدم تحديد عدد عناصر المصفوفة :
مثال : int A[m] ;
وعناصر المصفوفة تكون متجاورة وكل عنصر يحجز 2byte

A[0]	A[1]	A[2]	A[3]	A[4]

تمرين : [1] -
باستخدام المصفوفات اكتب برنامج يقوم بالتالي :
١- ادخال عناصر مصفوفة ذات بعد واحد .
٢- حساب مجموع عناصر المصفوفة :
٣- ضرب عناصر المصفوفة بشكل صحيح .
تمرين [2] -
اكتب برنامج يسمح بإدخال وطباعة عناصر مصفوفة :

```
# include <iostream.h>
# defin  n  5
Void main()
{
int  A[n],i ;
:  //  ادخال عناصر المصفوفة
for (i=0; i<n; ++i)
```

```
cin>>A[i];
for(i=0; i<n; ++i)
cout<<A[i];
}
```

للتوضيح أكثر:

```
for(i=0; i<n; ++i);
cout<<"\nA["<<i<<"]=";
cin>>A[i]
}
A[0]=5
A[1]=8
```

حل التمرين (1)

```
#include <iostream.h>
#define n 5
void main()
{
int A[n], B[n], k, s, i;
for(i=0; i<n; ++i)
cin>>A[i];
s=0;
for(i=0; i<n; ++i)
s=s+A[i]; cout<<"\ns="<<s
cout<<"\nk="; cin>>k;
for(i=0; i<n; ++i)
{
B(i)=k*A[i];
cout<<"\nB["<<i<<"]="<<B[i];
}
}
```

ملاحظة :

يمكن في الواقع تعريف مصفوفة واعطاء لها قيم ابتدائية مباشرة بدل حلقة For

مثال :

```
int A[3] = {8, 7, 13};
```

- المصفوفة ذات البعدين : وصيغتها :

;[دليل ثاني] [دليل اول] اسم المصفوفة — نوع متغيرات

مثال :

```
int A[3][4];
```

عدد الاسطر ←   → عدد الاعمدة

هنا لدينا مصفوفة مكونة من (3) اسطر (4) اعمدة .

الدليل الاول يمثل عدد الاسطر والدليل الثاني يمثل عدد الاعمدة .

- عناصر المصفوفة :

A[0][0]	A[0][1]	A[0][2]	A[0][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[2][0]	A[2][1]	A[2][2]	A[2][3]

تمرين :

باستخدام مصفوفة ذات البعدين اكتب برنامج يسمح ب

1- ادخال عناصر مصفوفة ذات بعدين .

2- طباعة هذه العناصر .

3- طباعة منقول المصفوفة (استبدال الاسطر بالاعمدة والاعمدة بالاسطر) .

4- طباعة عناصر القطر الرئيسي وعددها .

الحل :

1)
```
#Include <iostream.h>
# defin n 3
Vold main()
{
int A[n][n], i, j, k=0;
```

عداد حساب عدد عناصر القطر الرئيسي .

```
for(i=0 ; i<n ; ++i)
for(j=0; j<n; ++j)
  cin >> A[i][j];
```
إدخال

2).-
```
for(i=0; i<n ; ++i)
for(j=0; j<n; ++j)
 cout << A[i][j];
```
3).-
```
for (i=0 ; i<n ; ++i)
for (j=0 ; j<n ; ++j)
cout << A[j][i];
```
4).
```
for(i=0; i<n ; ++i)
for(i=0 ; j<n ; ++j)
  if (i==j )
  {
  cout << A[i][j];
  k=k+1 ;
  }
  cout<< "\n k = "<<k;
  }
```

ملاحظة:

- حساب مجموع عناصر القطر الرئيسي `S = S + A[i][j]`
- استبدل عناصر القطر الرئيسي بالعدد 25.
  `cout<<A[i][j]; cout<<25;`

- التعامل مع المتغيرات الحرفية والسلاسل الحرفية.

1. المتغيرات الحرفية char

وتتضمن الأحرف الكبيرة والصغيرة والرموز والإشارات وصيغتها:

`char اسم متغير;`

مثال `char c;`

عند اعطاء قيمة لمتغير حرفي يجب حصر القيمة بين اشارتي تنصيص مفردتين

مثال:

```
c='A';
cout<<c;        A
```

رمز فقط يظهر

بواسطة امر الطباعة cout عند طباعة الرمز أو طباعة رقم الترتيب

مثال:

```
int c;                      char c;
c='A';                      c='A';
cout<<c;   65               cout<<c;   A
```

تمرين: (وظيفة):

اكتب برنامج يطبع جدول الأسكي كود

مثال:

اكتب برنامج يطبع الأحرف الكبيرة: A , B , ... , Z

الحل:

```
#include <iostream.h>
void main()
{
char i;
for(i='A'; i<='Z'; ++i)
cout<<i;
}
```

- طباعة ارقام الترتيب نكتب بدل char i; نكتب int i;

تمرين (وظيفة):

اكتب برنامج يطبع الأحرف الكبيرة وارقام الترتيب.

تمرين: اكتب برنامج يسمح بإدخال نص من لوحة المطابيع حرف حرف ينتهي ادخال الأحرف بالضغط على اشارة (*) وطباعة عدد أحرف النص المدخل واطبع الاحرف المدخلة.

الحل:

```
#include <iostream.h>
void main()
{
char c;
int i=0;
do
{
cin >>c;     cout << c;
i = i+1;
}
while (c!='*');
cout <<"\ni =" << i;
}
```

طباعة: cout << c;

- التعامل مع السلاسل الحرفية:

هي عبارة عن سلسلة او تتابع خطي من الرموز او الاحرف والارقام والاشارات ... الخ
نوع المتغيرات char يستخدم من اجل التعامل مع المتغيرات الحرفية. اضافة الى ذلك
نوع المتغيرات char يستخدم من اجل التعامل مع السلاسل الحرفية وهنا يجب
ان نحدد طول السلسلة الحرفية عند تعريف هذه السلسلة وصيغتها:

char اسم السلسلة [طول السلسلة] ;

مثال:

```
char s[15];
```

هنا لدينا سلسلة حرفية اسمها s مكونة من (15) حرف.
وعند اعطاء قيمة للسلسلة الحرفية يجب وضع هذه القيمة بين اشارتي تنصيص مزدوجة
مثال:

```
s= "Borland";
```

يمكن ايضا تعريف السلسلة واعطائها قيمة:
مثال:

```
char s[6] = "basic";
```

عند التعامل مع السلسلة الحرفية يمكن في الواقع التعامل مع السلسلة الحرفية حرف حرف او التعامل مع السلسلة الحرفية اسم .

مثال :

```
char s[6] = {'b','a','s','i','c','\0'}
```

(\0 الحرف الصفري)

تم عادة الحاق الحرف الصفري في نهاية السلسلة وهو يعبر عن نهاية السلسلة . ولا يظهر في الطباعة

من قبل مترجم اللغة

لذلك يجب ان نحجز موضع للحرف الصفري .

ـ يمكن ايضاً تعريف السلسلة الحرفية كاسم او سلسلة ويجب حصر الاسم بين اشارتي تنصيص مزدوجة

مثال :

```
char s[6] = "basic";
```

لادخال السلسلة الحرفية حرف حرف نستخدم حلقة For اما الادخال السلسلة الحرفية كاسم من لوحة المفاتيح نستخدم الامر :

```
cin >> s;
```

اسم السلسلة

مثال :

```
char st[20];
cin >> st;
```

ـ عند ادخال السلسلة من لوحة المفاتيح نُدخل السلسلة اسم من دون وجود الفراغ لان الفراغ يمثل فاصل بين السلسلتين .

ـ وعند ادخال الاسم او السلسلة الحرفية من لوحة المفاتيح مع وجود الفراغ نستخدم الامر :

```
gets (st);
```

وحتى نستخدم الامر gets نستخدم الامر :

```
#include <stdio.h>
```

ـ ولطباعة السلسلة الحرفية نستخدم الامر cout

مثال :

`cout << st ;`

- لطباعة السلسلة حرف حرف نحتاج إلى حلقة : for

هناك بعض الدوال التي تستخدم مع المتغيرات الحرفية :

1 - الدالة (متغير) toupper

تحويل الحرف الصغير الى حرف كبير .

2 - الدالة (رمز) to Lower

تحويل الحرف الكبير الى حرف صغير .

حتى نستخدم جميع الأوامر نستخدم الأمر :

`#include <ctype.h>`

مثال :

```
# include <ctype.h>
char c='d';
char A='G';
    toupper (c);      D
    to Lower (A);     g
```

- ويوجد ايضاً الملف الرأسي

`#include <string.h>`

للتعامل مع السلاسل الحرفية وهناك بعض الاوامر في السلاسل الحرفية وهي :

1 - `strcpy (s1, s2) ;` (s1 : متغير ، s2 : قيمة)

وتعني نسخ قيمة لسلسلة حرفية الى متغير .

مثال :

```
char A[20];
char B[10] = "Borland";
```

سيتم نسخ Borland الى S_2 (اي نسخ قيمة الى متغير .

2 - `strcomp (s1, s2);`

المقارنة بين سلسلتين وتوقف عملية المقارنة عند اول حرفين غير متطابقين .

وتعطي قيمة موجبة إذا كان $S_1 > S_2$

وقيمة سالبة إذا كان $S_1 < S_2$

قيمة صفر $S_1 = S_2$

3- strcat (S1, S2) ;

تستخدم من أجل إلحاق السلسلة الثانية في نهاية السلسلة الأولى ويجب أن تكون السلسلة الأولى كبيرة بقدر كاف حتى تتسع للسلسلتين.

مثال:

```
char A(70) = " borland" ;
char B (10) = " c++ " ;
```

4- strupr (st) تحويل سلسلة لأحرف كبيرة.

5- strlwr (st) تحويل سلسلة لأحرف صغيرة.

6- مهم: x = strlen(s) ;

تستخدم من أجل أن تعطينا طول السلسلة (عدد الأحرف)

وهذه الدوال تستخدم مع الأمر (الملف الرئيسي):

```
#include <string.h>
```

تمرين: اكتب برنامج يعرف سلسلة حرفية وإعطاء قيمة لهذه السلسلة ثم اطبع هذه السلسلة:

1- طباعة السلسلة اسم كامل.

2- طباعة السلسلة حرف حرف.

انتهت المحاضرة

٢٨